



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 150  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/676,373

09/30/2003

Stefan Jesse

09700.0216-00

3224

60668 7590 04/09/2007  
SAP / FINNEGAN, HENDERSON LLP  
901 NEW YORK AVENUE, NW  
WASHINGTON, DC 20001-4413

EXAMINER

VU, TUAN A

ART UNIT

PAPER NUMBER

2193

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
--	-----------	---------------

3 MONTHS

04/09/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

<b>Office Action Summary</b>	Application No.	Applicant(s)	
	10/676,373	JESSE ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 1/26/07.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-22 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-22 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                       | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>10/19/06</u> .  | 6) <input type="checkbox"/> Other: _____                          |

### DETAILED ACTION

1. This action is responsive to the Applicant's response filed 1/26/07.

As indicated in Applicant's response, claims 7-8, 11, 20-22 have been amended. Claims 1-22 are pending in the office action.

#### *Double Patenting*

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 8, 18 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 4, 12, 19 of copending Application No. 10,676,374 (hereinafter '374).

Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following example of conflicting claims.

As per instant claim 8, copending '374 claim 4 recites a first data model being used to derive an API and employing the API to access development objects. But '374 claim 4 does not explicitly recite (i) object model to structure development objects; (ii) generate intermediate objects therefrom and (iii) base on the set of intermediate objects and a code template to generate the API, the model including (iv) a language extension used to implement a feature of the API such as an indication of a file border.

However, '374 claim 4 recites a variation of the language teaching limitations (i) and (ii) via the recital of 'defining file borders comprising identifying of development objects to be included in a file ... in the data model ... to be children of the main ... object that are not identified as main...objects', the intermediate objects being added objects to the file of the main object including development objects defined in the data model. As for the template code (iii), '374 claim 4 includes file storing user-defined code associated with the main development object; and for one skill in the art, having a template file (see '374 claim 4) for user to define code for a development as purported by '374 claim 3 would be equivalent to (iii), thus disclosed or otherwise obvious. As for the feature extension comprising a file border indication referred to as (iv), this is suggested in '374 reciting of 'defining file borders for development', and storing development objects in a repository based on the file borders, and accessing these objects via the API (\*); so that one skill in the art would be motivated to provide an extension structure obtained from the repository (e.g. template builder) in the course of the API derivation with utilizing of information in the '374 stored file-based repository for the derivation. That is, the information thus extended (e.g. via a template builder) from the stored model/repository regarding a particular file border identity would be used to support the creation of API parameter or

Art Unit: 2193

attributes which would be needed to access the very components stored from the '374 defining of file borders, as purported by the endeavor described as (\*) from the above.

**As per instant claim 18**, '374 claims 12 and 19 also recite an analogous language expressing receiving in a development method a data model (being generated, repository storing development in a data model), deriving a API based thereon; and use the API to perform operations on a development object (e.g. API incorporating the feature defined by the model customizable extension during development of the application; OR API to access development objects).

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-22 are rejected under 35 U.S.C. 102(e) as being anticipated by Ho et al., USPN: 6,964,053 (hereinafter Ho).

**As per claim 1**, Ho discloses a computer program product, tangibly embodied in an information carrier, for developing an application, the computer program product being operable to cause data processing apparatus to:

Art Unit: 2193

receive a first data model in a first language (*Rose* documents – col. 10, lines 19-29; *Rose* File 601- Fig. 6; input HTML form – col. 13, lines 21-27), the data model being used to structure development objects;

generate a set of intermediate objects based on the first data model (e.g. DTD, schema 605 – Fig. 6; Fig. 8; *metamodel files*, *DTD files* col. 11, lines 56 –58 ; IMS XML-formatted *messages*); and

based on the set of intermediate objects and a code template (e.g. Fig. 4; Fig. 7-8; *template* - col. 15, lines 44 to col. 16, line 26; XML... messages – col. 13, lines 21-34; Fig. 9-10), generate an API to access the development objects (e.g. col. 12, lines 48 to col. 13, line 8; Fig. 3-4; *connector* - col. 9, lines 25-29; *API* -col. 13, lines 9-20; col. 14, lines 29-67 – Note: metamodel descriptor or class signatures metadata --representing template definition of a class-- being based upon in the main interface tool --- see Application interface 707, Fig. 7 -- for retrieving objects via triggering invocation of a Connector or MQ API **reads on** intermediate objects; and this connector API reads on creating a API – see *interface 705*  $\leftrightarrow$  *connector 701*, Fig. 7; *connector ...then calls ...application API* - col. 13, li. 1-15; col. 15, lines 1-20 – to access the objects defined as metamodel type code template inside intermediate OTMA messages whose inputs/outputs are processed – col. 15, lines 1-20).

**As per claims 2-4**, Ho discloses instructions to convert the first data model to a second data model in a second language, wherein the set of intermediate objects is based on the second data model (e.g. *generate a DTD ... XML documents of the application source files* - col. 11, lines 13-24; *DTD, schema 605* – Fig. 6; col. 11, lines 56 –58; col. 15, lines 1-20); wherein the second language comprises XML and the first language comprises UML (col. 11, lines 13-24).

**As per claim 5**, Ho discloses wherein the set of intermediate objects comprises Java objects (e.g. Java code 609 – Fig. 6; Fig. 8-19 ).

**As per claims 6-8**, Ho discloses wherein the first language comprises a customizable extension; wherein the customizable extension is used to implement an additional feature of the API, wherein the additional feature comprises an indication of a file border (Fig. 9-10; col. 14, line 58 to col. 15 line 42 – Note: customizing of interface to fulfill requests for data translation based on runtime read of *metamodel files* based on construct neutral structures to facilitate language translation – see col. 15, lines 21-29-- to appropriate platform reads on customizable runtime extension; and type descriptor files to support Java, cobol or IMS to enable customization of connector implementation read on model extension – see Fig. 3-4 – with indication pointing to file borders – or descriptor files).

**As per claim 9**, Ho discloses wherein the API comprises a copy and paste operation (e.g. Fig. 2; Fig. 16-17 – Note: customization from user interface to create instance of metamodel connector to comply to a specific platform request reads on editing capabilities of developer – see *Integration runtime 221*, Fig. 2; while requests based on said created connector or SAX – see col. 9, lines 13-29 – being real time event-based should read on user capability of modifying screen content – see col. 37, lines 11-39; col. 38, lines 8-65, hence copy and paste).

**As per claim 10**, Ho discloses a computer program product, tangibly embodied in an information carrier, for developing an application, the computer program product being operable to cause data processing apparatus to:

receive a first data model in a first language, the data model being used to structure development objects (e.g. *Rose* documents – col. 10, lines 19-29; *Rose File 601*- Fig. 6);

generate a set of intermediate objects based on the first data model; and based on the set of intermediate objects and a schema template (e.g. source 507 – Fig. 5; *instance file 613* -col. 11, lines 13-24; Fig. 4; *XMI instance* - Fig. 6, 8 – Note: XMI instance file reads on intermediate objects and template leading to constructing a DTD or XML schema),

generate an XML schema (e.g. *DTD, schema 605* – Fig. 6; col. 11, lines 56 –58; XML 511, Fig. 5) used to implement the development objects.

**As per claims 11-14**, refer to the corresponding rejection as set forth in claims 2-5.

**As per claims 15-16**, Ho discloses wherein the XML schema includes a tree based on aggregation relationships in the first data model; wherein the XML schema includes a reference based on an association relationship in the first data model (e.g. col. 11, lines 13-24, lines 56-58).

**As per claim 17**, Ho discloses wherein the XML schema includes a complex type extension based on an inheritance relationship in the first data model (e.g. Fig. 3-5; Fig. 6; *TypeDescriptor Metamodel* Fig. 9-10; *inheritance* -col. 15, lines 43-67; col. 23, lines 6-54 – Note: creating of schema for repository of Rose files to identify typedescriptor metamodel files reads in schema including indication pointing to complex type extension – see cols. 14-16 for runtime marshalling/transformation as purported in Fig. 3 integration environment).

**As per claim 18**, Ho discloses a computer program product, tangibly embodied in an information carrier, for developing an application, the computer program product being operable to cause data processing apparatus to:

receive a first data model;

derive an API based on the data model; and

use the API to perform operations on a development object;



Art Unit: 2193

all of which limitations having been addressed in claim 1.

**As per claim 19**, Ho discloses wherein the API (Fig. 3-4) comprises an interface layer, a proxy layer (e.g. API, proxy – col. 9, lines 13-42), and a state layer (e.g. *SimpleInstanceTD* – Fig. 10B; *InstanceTDBase* – Fig. 12-13).

**As per claim 20**, Ho discloses wherein the operations comprise creating a new development object as a transient object (e.g. Interface Metamodel 707 - Fig. 6-7; Figs. 12-13 and related instantiation of objects for a generic model placeholder – see col. 23, lines 6-54) without an existing corresponding file; and modifying the transient object until the transient object is committed to a persistent file (e.g. *Upperbounds*, *Alignment requirements*, *Platform compiler Type*, *validity checking* -cols 24-36; Fig. 16 – Note: validity checking or bounds checking for instance of Enumeration objects reads on modifying until an object is ready for being compiled into target platform).

**As per claims 21-22**, Ho discloses comprising instructions to destroy the transient object if a delete command is requested before the transient object is committed to a persistent file; and to mark the persistent file as deleted if a delete is requested after the transient object is committed to a persistent file (e.g. *commit* – col. 13, lines 60 to col. 14, line 9; ...*returned to the manufacturer's server* - col. 37, lines 4-57; *track changes*, *mark annotations* – col. 38, lines 19-65 – Note: complex operations involving data invocations via servers, suppliers or manufacturing sites with a *commit* for any received messages or not accepting incompatible set of (platform) data at the user editing interface – or conference/authoring level - **read on** transient object marked for delete before it gets persisted at manufacturing site or vendor supply chain).

### ***Response to Arguments***

Art Unit: 2193

6. Applicant's arguments filed 1/26/07 have been fully considered but they are not persuasive. Following are the Examiner's observations in regard thereto.

**Rejections under 35 USC § 102(e):**

(A) As per claims 1-9, Applicants have submitted that the DTD, XML schemas or Cobol metamodel are proffered in the Office Action as set of intermediate objects to the first data model proffered as Rose files; and this does not disclose 'based on the set of intermediate objects and a code template ... generate an API to access the metamodels' ( Appl. Rmrks pg. 8, bottom; pg. 9, first half). The rejection has interpreted claim 1 as a scenario having 3 steps: (i) receive a first data model in a first language; (ii) generate intermediate objects based thereupon; (iii) generate API for accessing these objects based on those objects and some code template.

The rejection has pointed to sections of Ho to address

(i); that is, Ho's teaching about receiving of Rose files and source files to create a Metamodel to derive corresponding schemas; or to read in a HTML request input form (see col. 13);

(ii), that is, deriving the schemas from the Rose models and source files (step 605 - Fig. 6) or IMS formatted messages (col. 13); and

(iii); that is, to use the main Application metamodel (Fig. 7) including *meta-definition messages* to trigger additional connectors leading to corresponding APIs to retrieve definition of objects represented by the metamodel ( see MQ API - col. 12 bottom, R to col. 13 top); or processing the requirements of a Metamodel type Descriptor inside each IMS OTMA messages to trigger invocation of OTMA connector ( e.g. Fig. 7; col. 15, lines 1-20) wherein each connector acts as an application interface invocation (see col. 9, lines 25-29), and wherein each

Art Unit: 2193

metamodel type Descriptor includes a class signature definition – or template - while parsing the input or outputs messages – see Fig. 9-10 -- in processing the content of the IMS meta-messages being interchanged (with the main MetaModel tool) and while validating OO objects (as marshalling unmarshalling scheme) while using the main connector-based tool (see col. 15-18); i.e. based on the intermediate messages to retrieve more definition or data structures therein ( see OTMA - col. 13) using connector being invoked or triggered as APIs to obtain meta-definition in the parsing/validating process.

It is deemed that the rejection has addressed the limitations as presented in the claims in light of interpretation of what has been recited. The argument is therefore not persuasive.

(B) Applicants have submitted that Ho's connector is not a API to access developments objects ( Appl. Rmrks pg. 9, bottom half), not based on the set of intermediate objects, but rather in terms thereof (Appl. Rmrks pg. 10, top). The claim does not provide differentiating details between what is questioned here as 'based on' and 'in terms of', hence this argument does not amount to sufficient grounds as to why Ho's connector is precluded from being a application interface being generated based on the processing of meta-data contained in OTMA messages as these are successively created and intercommunicated between the main Metamodel tool and middleware support to address requirements of a object-oriented class signature – code template. The argument is hence not persuasive, notwithstanding the analysis set forth in section A.

(C) Applicants have submitted that IMS OTMA metamodel is not an API to access and that IMS transaction message are only for input output request ( Appl. Rmrks pg. 10, 2<sup>nd</sup>, 3<sup>rd</sup> para). The claim language as recited has been interpreted and addressed in light of the analysis in section A above. This argument is also referred thereto, because the argument that a message is

Art Unit: 2193

about input/output request is not commensurate with the actual claim language as it has been interpreted (refer to section A). Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(D) As per claims 10-17, Applicants have submitted that Ho fails to teach 'based on the set of intermediate objects and a code template ... generate an XML schema used to implement ... objects' and that Ho's XMI as asserted by the Examiner cannot support meeting claim 10 (Appl. Rmrks pg. 11, middle). The rejection has set forth a scenario in which variety of model files are read into a Importer tool framework, such files including Rose, BMS, Cobol, PL/I, MFS files (see Fig. 6) and use the code template (e.g. format according to W3C technology) provided by the XMI file being generated as intermediate objects to create a corresponding DTD or XML documents. The argument on the terms 'based on' is not convincing (refer to section B) and as a whole the argument remains non-persuasive because the claim lacks specificity so as to support the teaching mentioned by the argument. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(E) For claims 18-22, Applicants have submitted Ho's API does not perform operations on development objects; and nowhere does Ho teach an API comprising a stub (Appl. Rmrks pg. 12). The limitation referred as 'use the API to perform on a development object' using a API has been interpreted as operations performed on the objects are such that they are direct result from

Art Unit: 2193

using this API. When the claim does not explicitly disclose what 'performed on the development objects' actually amounts to, it is reasonable to interpret this in a broad manner. When development objects are retrieved by means of connector via MQ API or database queries remote calls, such retrieval can be analogized as being performed on the stored data in a way that the data has to be marshaled in a message form to be sent back to the requesting calls. Claim 1 has addressed the API via invoking connector to retrieve OTMA persisted meta-definitions and validate the meta-Descriptor inside these communicated IMS-formatted messages to implement more code. API performing action of formatted the meta-data into messages form being communicated to the runtime tool (see Fig. 2, 7) has been disclosed. The argument is deemed not persuasive to preclude the MQ API by HO from reading into this 'perform operations' limitation. The interchange CAM model by Ho teaches marshalling of data ( col. 10, lines 40-41) via using of messaging and connector invocations. Besides, the term stub is not part of the claim language. The argument is being misplaced because it is not commensurate with the very language of the claim.

(F) Applicants have submitted that HO's API does not comprise a state layer (Appl. Rmrks pg. 13). How much the 'comprises' limitation is taught in the claim is yet left to one of ordinary skill in the art's broad interpretation. How much the 'state layer' amounts to in function of the above *comprises* broad connotation is also left to broad interpretation. If an application program interface has layer, the underlying or explicit specifics as to why it has to be layered is not evident from the claim. Normally an API does not get implemented by having layered hierarchies as in a network or computer organization; hence the term layer would be treated as a mere abstract entity or structure. Any API when it is invoked does include a programmatic state;

Art Unit: 2193

and the InstanceTD example provided in the rejection is one of those entities being created during the XMI processing for instantiation at runtime leading to the marshaling of a request (see col. 18), all of which linking object instance dictating a compiler state leading to linking error. That is, when a message being invoked to retrieve data for a marshalling process, the compiler state of a InstanceTD amounts to this state layer inside the API at the source of the MQ API or OTMA message interchange using the XMI parsing. Notwithstanding the fact that the limitation has been recited with very broad concepts, the argument is not convincing. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Based on the above, the claims will stand rejected as set forth in the Office Action.

### *Conclusion*

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2193

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 ( for non-official correspondence - please consult Examiner before using) or 571-273-8300 ( for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu  
Patent Examiner,  
Art Unit 2193  
April 02, 2007